

FACULDADE DE VIÇOSA
MOSTRA ACADEMICA 2009

MoldesView: uma ferramenta para visualização gráfica de soluções em problemas de corte e empacotamento.

Rodrigo César Gamarano Mariano¹

Ariana Priscila da Silva²

Manoel Vieira²

Robson de Sá Rodrigues²

Heleno do Nascimento Santos³

RESUMO

A apresentação dos resultados gerados por modelos de corte e empacotamento, normalmente utilizada sem auxílio de visualização gráfica, dificulta bastante ao usuário a interpretação da solução obtida. Permitir que sejam visualizadas as localizações dos polígonos encaixados em uma determinada área, verificar a ocorrência de sobreposição destes itens (figuras), ter uma noção mais clara das características geométricas de tais itens, além da proporção de encaixe dos polígonos em relação a sua área de ocupação, saber a localização das áreas relativas a perda, dentre outras, são características extremamente desejáveis em sistemas que contemplam este tipo de modelo. Com base nos requisitos do cenário apresentado, foi desenvolvida uma ferramenta de auxílio, a qual denominou-se MoldesView, para a visualização de resultados de modelos relacionados ao problema de corte e empacotamento, que poderá ser aplicada, por exemplo, no ramo de confecção de roupas. Nesta ferramenta são tratadas soluções para o problema de corte e empacotamento para dimensões unidimensional e bidimensional. Durante seu desenvolvimento, levou-se em consideração teorias relacionadas à Programação Inteira, princípios de desenvolvimento em Delphi e, conseqüentemente, uso da classe TCanvas. Com a MoldesView o usuário pode informar a cor e as dimensões relativas ao tecido, importar e visualizar as figuras que serão utilizadas no empacotamento, visualizar os moldes já dispostos sobre o tecido, visualizar o encaixe peça a peça, podendo ainda, exibir o percentual de perdas e de ocupação dos itens.

¹ Graduando em Sistema de Informação (FDV) e bolsista do programa BITEC.

² Ifour Sistemas e Bolsistas da FAPEMIG.

³ Faculdade de Viçosa

1. INTRODUÇÃO

Problemas de corte têm sido amplamente estudados e discutidos por pesquisadores, devido à grande importância econômica de sua aplicação em diversos ramos da atividade humana, como a indústria de tecidos, de calçados e de couros.

Especificamente, para o caso do Problema de Corte e Empacotamento, a captura de padrões de cortes, representados pelas peças, e o arranjo dessas peças em planos de corte viáveis constituem um desafio, pois este problema é encontrado em vários setores industriais.

Normalmente, o objetivo para o problema de corte é minimizar o desperdício do material envolvido no processo, tornando o produto final mais competitivo no mercado, diminuindo custos envolvidos na aquisição da matéria prima.

Segundo CAMPOS, (2007), o problema de corte é comumente encontrado nas indústrias em que a produção é baseada na extração (corte) de peças, a partir de um plano (matéria-prima) de tamanho maior, como indústrias de móveis, vidros, metais, confecções, couros, espumas, papelões e outras. Devido à variação de formas, tamanhos e de demandas para cada tipo de peça, qualquer arranjo proposto para corte gera sobras e espaços sobre o plano material, entre uma peça e outra, o que, evidentemente, implica em desperdício.

Segundo MORABITO E BELLUZZO (2007), a complexidade do problema de corte é classificada como NP-Difícil, ou seja, não existe um algoritmo com tempo de execução polinomial que possa obter uma solução ótima. Dessa forma, as soluções se baseiam em heurísticas, que são procedimentos que não geram solução exata, mas que visam aproximar ao máximo da exatidão, gerando resultados ótimos em tempos de execução aceitáveis. Desta forma, visando a aplicação destas heurísticas, são criados e utilizados otimizadores que retornam soluções próximas a ótima para o usuário, mas que, em determinadas situações, podem obter soluções bastante razoáveis, que poderiam ser melhoradas através da interação do otimizador com o usuário.

A solução de problemas de otimização ou de simulação, vinculados ao Problema de Corte e Empacotamento, requer a interação entre o usuário e as soluções geradas, a fim de garantir a troca de informações necessária para alcançar um resultado de forma mais eficiente e eficaz. Assim, uma ferramenta que possa auxiliar a visualização gráfica das soluções obtidas, para este tipo de problema, seria uma alternativa para iniciar este processo iterativo, garantindo que o usuário visualize passo a passo como o problema está sendo solucionado pelo otimizador e permitindo que ele descubra problemas relativos a sua forma de funcionamento.

A ferramenta na qual denominamos MoldesView, será uma interface entre o usuário e o sistema Gerador de Plano de Corte, que é o núcleo de um sistema de informação que está sendo desenvolvido para resolver um problema de corte encontrado, por exemplo, na indústria de confecções.

2. OBJETIVOS

O objetivo geral desse projeto é a geração de um visualizador gráfico que permita acompanhar a ocupação de uma peça de tecido por um conjunto de peças componentes de vestuário (calças, camisas, etc.), de acordo com a solução gerada pelo otimizador de encaixes.

Especificamente, pretende-se:

- Permitir a visualização gráfica do encaixe de peças uma a uma;

- Contribuir, de forma visual, com a fase de testes do otimizador de encaixes;
- Visualizar a estrutura completa do encaixe associado a uma solução de um problema de corte e empacotamento.

3. METODOLOGIA

A metodologia foi dividida da seguinte forma:

- Fundamentação teórica, na qual foi revisado o conceito de Programação Inteira e Problemas de Corte e Empacotamento.
- Desenvolvimento do sistema, qual foi usada o compilador Delphi, com uso da classe TCanvas.

3.1 FUDAMENTAÇÃO TEÓRICA

3.1.1 PROGRAMAÇÃO INTEIRA

Segundo LUNA E GOLDBARG (2005), a Programação Inteira ocorre quando as variáveis do problema pertencem ao conjunto de números inteiros, mas há casos em que as variáveis são do tipo mistas, sendo uma parte do tipo inteiro enquanto outras do tipo real.

Em muitos problemas práticos, variáveis de decisão só fazem sentido se forem inteiras. Por exemplo:

- Números de computadores a serem comprados para uma organização.
- Números de funcionários a serem contratados, para produzir número de itens para o setor industrial.

Pode-se observar que tais soluções dos problemas citados são inteiras, pois não se pode comprar 50,5 computadores ou até mesmo contratar 200,5 novos funcionários.

As aplicações da programação inteira também ocorrem quando é necessário tomar decisão do tipo sim ou não como, por exemplo:

- Devemos iniciar determinado projeto?
- Devemos construir uma nova fábrica em determinado local?

Para essas decisões, podem-se utilizar variáveis inteiras binárias, tais:

- $X_j = 0$, se a decisão j for **Não**.
- $X_j = 1$, se a decisão for **Sim**.

3.1.2 PROBLEMAS DE CORTE E EMPACOTAMENTO

Segundo MORABITO E BELLUZO (2007), os problemas de corte e empacotamento, caracterizam-se por serem de problemas de otimização combinatória que envolve determinar o arranjo ótimo de unidades menores (itens) dentro de unidades maiores (objetos).

Vários sistemas de produção contêm processos de alocação desses itens, como é o caso do MODA01, que atende a área de confecção. Porém, existem aplicações outras, como o corte de bobina de papel, barra de aço, chapas de vidros, madeiras, couro e plástico, que se assemelham bastante com o problema tratado no MODA01.

Segundo FRACESCHINELLI (2002), o modelo matemático que representa tais problemas é de programação linear inteira e o grande número de variáveis envolvidas dificulta sua resolução.

3.2 DESENVOLVIMENTO

Utilizou-se para a codificação a ambiente Delphi, versão 7. A escolha desta linguagem se deu pelo fato de ser uma linguagem já conhecida, garantindo a redução da curva de aprendizado para o desenvolvimento do projeto, sendo bem aceita pelos desenvolvedores de sistemas. A plataforma Windows foi utilizada para permitir a compatibilidade com o ambiente Delphi.

A classe TCanvas, do Delphi, permite que várias figuras geométricas sejam desenhadas: quadrados, triângulos, circunferências, dentre outras.

Alguns métodos pertencentes a esta classe podem ser destacados:

- Pixels – Propriedade do tipo Tcolor que indica a cor de cada um dos pontos (pixels) que formam o Canvas.

Exemplo: Para alterar a cor do ponto pertencente a coordenada (10,10) para azul, utiliza-se a sintaxe a seguir:

```
Image1.Canvas.Pixels[10,10] := clBlue;
```

- Pen – Propriedade do tipo Tpen determina o estilo da caneta que será utilizada para desenhar no Canvas (através dos métodos LineTo e MoveTo).

Exemplo: Para alterar a cor da caneta do Canvas e a espessura do componente Image1 para vermelho, utiliza-se:

```
Image1.Canvas.Pen.Color := clRed;
```

```
Image1.Canvas.Pen.Width := 3;
```

- Brush – Propriedade do tipo Tbrush que determina o estilo de preenchimento utilizado no Canvas.

Exemplo: Para alterar a cor do preenchimento para azul, faz-se:

```
Image1.Canvas.Brush.Color := clBlue;
```

- FillRect - Método utilizado para preencher uma porção retangular do Canvas pelo estilo especificado pelo Brush.

Exemplo: Para preencher um retângulo que possui vértices nos pontos (50,50) e (100,100):

```
Image1.Canvas.FillRect(Rect(50,50,100,100));
```

- MoveTo – Método utilizado para mover a caneta para um determinado ponto do Canvas. Este método apenas move a caneta, sem desenhar.

Exemplo: Image1.Canvas.MoveTo(100,100)

- LineTo – Método utilizado para traçar uma reta deste o ponto atual até o ponto especificado.

Exemplo: Para desenhar um triângulo de vértices (50,50) (50,100) e (100,100), pode ser digitado:

```
Image1.Canvas.MoveTo(50,50);
```

```
Image1.Canvas.LineTo(50,100);
```

```
Image1.Canvas.LineTo(100,100);
```

```
Image1.Canvas.LineTo(50,50);
```

- Font – Propriedade do tipo Tfont que determina como os textos serão escritos no Canvas (pelo método do TextOut, por exemplo).
- TextOut – Método que exibe uma texto no Canvas na coordenada especificada.

Exemplo: Image1.Canvas.TextOut(50,50,'Exemplo');

- Rectangle – Método que desenha um retângulo no Canvas. Devendo especificar as coordenadas do canto superior e do canto inferior direito.

Exemplo: Image1.Canvas.TextOut(50,50,100,100);

- Ellipse – Método que desenha uma elipse inscrita dentro de um retângulo imaginário.

Exemplo: Image1.Canvas.Ellipse(50,50,100,100);

O desenvolvimento do sistema foi dividido em algumas etapas.

Configuração inicial – Permite que o usuário forneça informações relativas ao tecido, dimensões e cor, essa atividade pode ser ativada através do formulário exibida na Figura1.

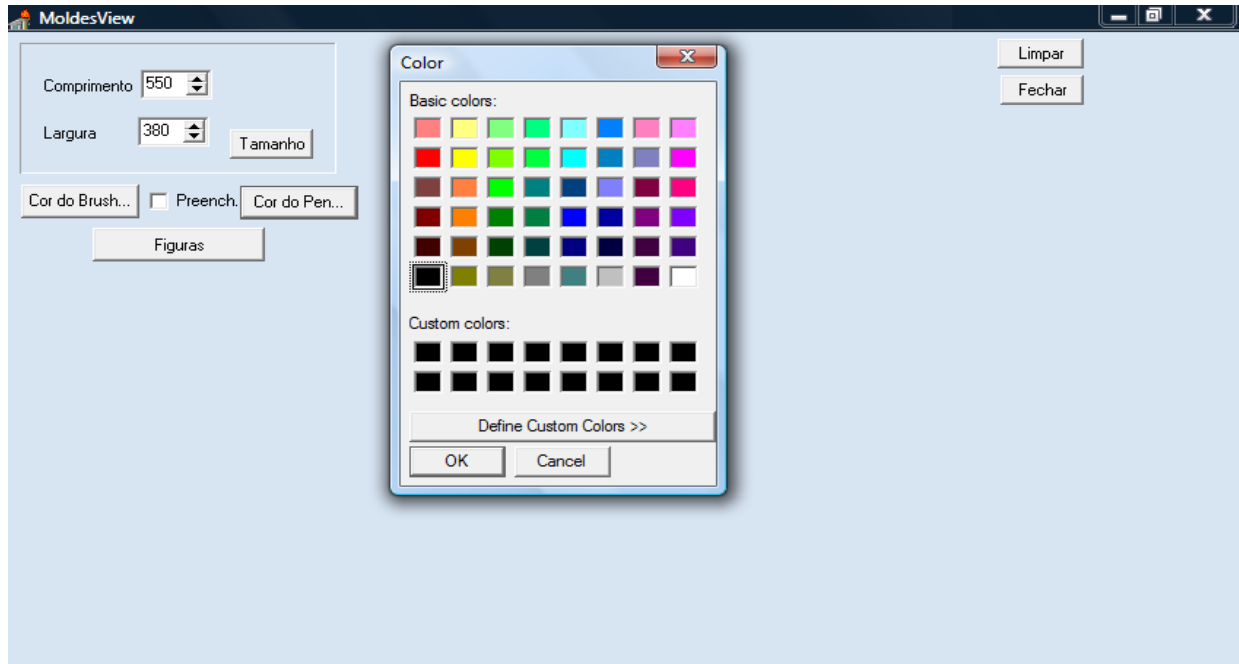


Figura1 - Tela de Configuração

- **Buscar os Itens (Figuras)**- Permite que o usuário informe quais arquivos possuem informações relativas às figuras, essa atividade pode ser ativada através do formulário exibido na Figura2.

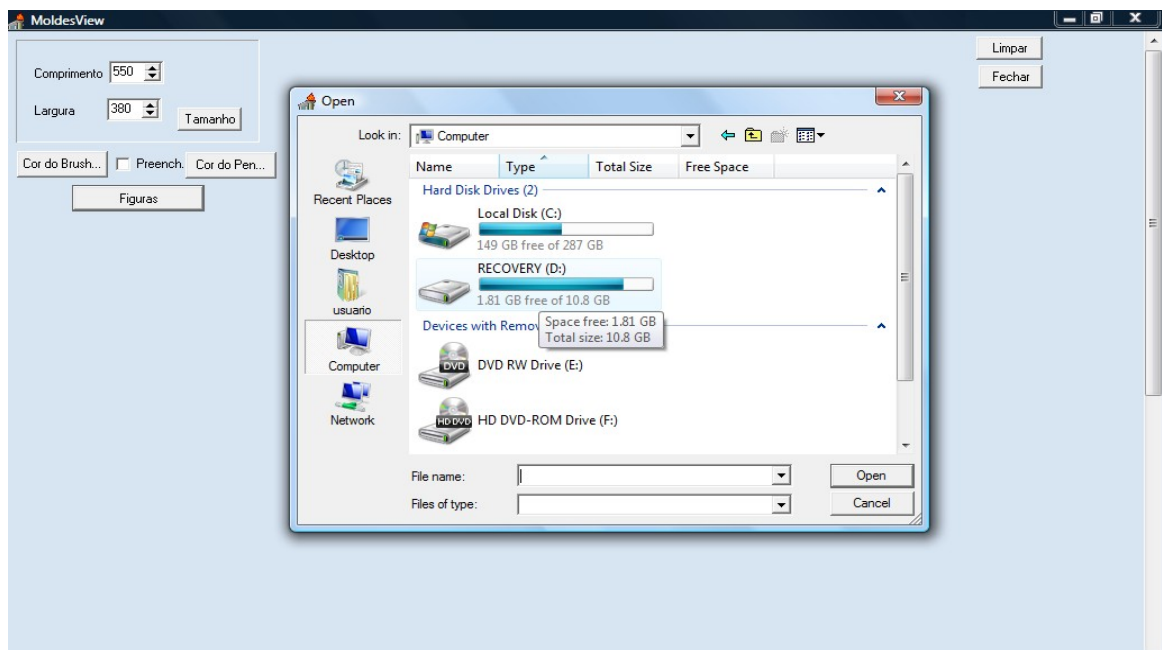


Figura2: Importação das Figuras

- **Visualização das Figuras**- Permite que o usuário visualize as figuras selecionadas, essa atividade pode ser ativada através do formulário exibido na figura3.

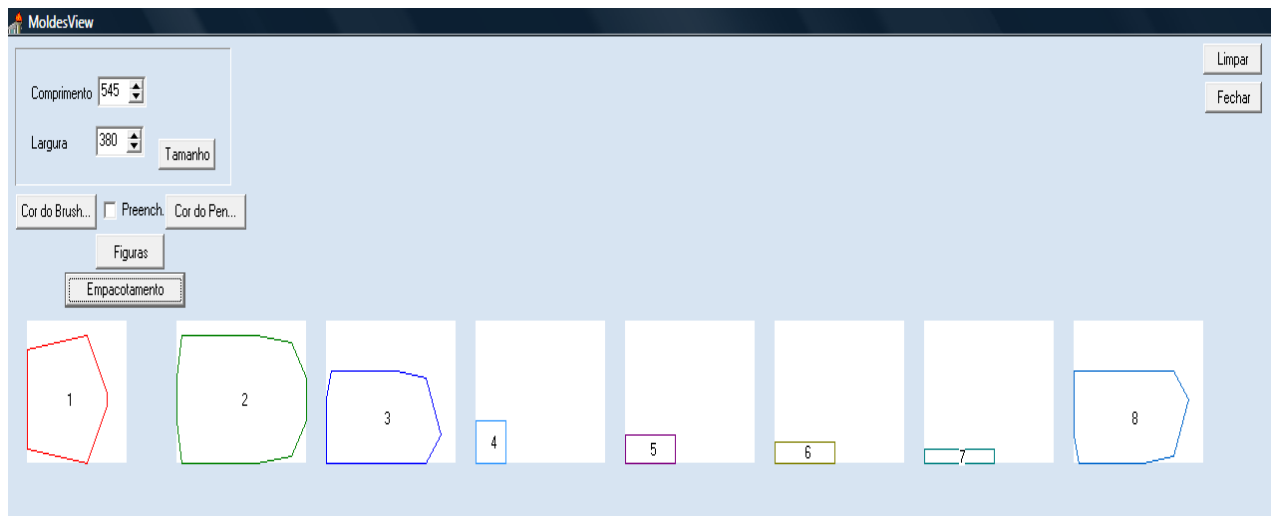


Figura3 - Visualização das Figuras

- **Visualizações Peça por Peça**- Permite que o usuário visualize a solução gerada pelo otimizador de forma mais detalhada, peça a peça, essa atividade pode ser ativada através do formulário exibido na Figura4.

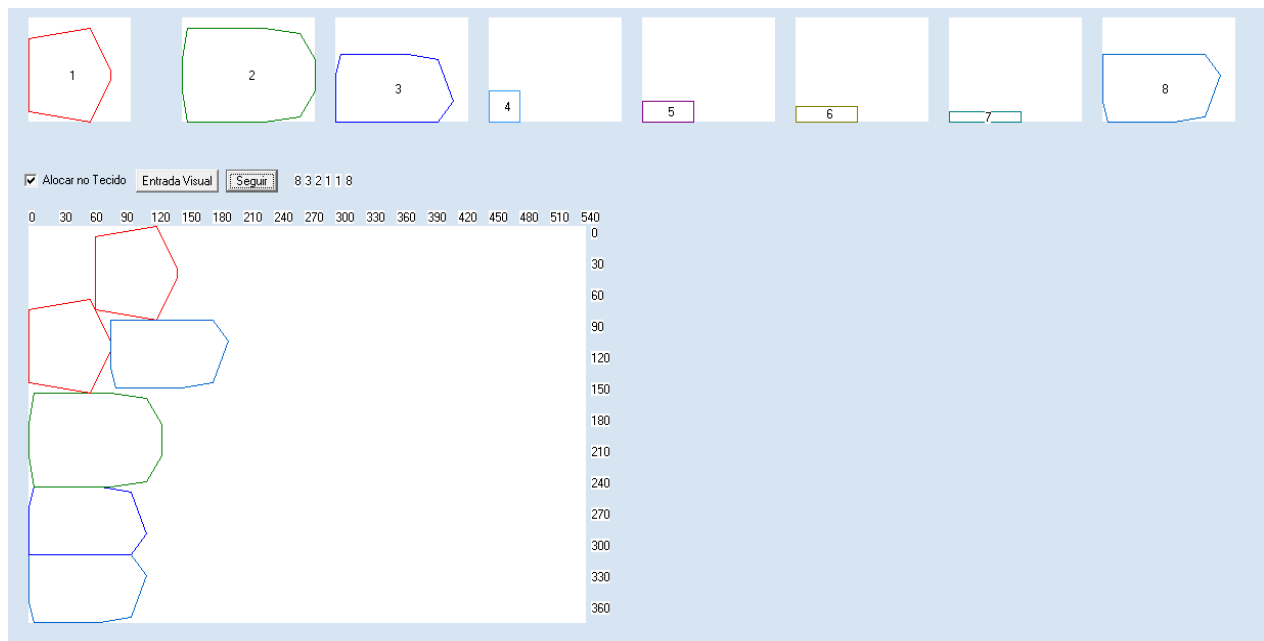


Figura4 - Entrada Peça por Peça

- **Visualizações dos Encaixes-** Permite que o usuário visualize a solução gerada pelo otimizador. Na qual demonstraremos na discussão dos resultados, através da Figura6.

4. RESULTADOS E DISCUSSÃO

Como exposto em objetivos, o principal resultado esperado com as atividades desenvolvidas é a construção de uma interface entre o usuário e o Sistema Gerador de Plano de Corte, que é o núcleo de um sistema de informação que está sendo desenvolvido para resolver um problema de corte encontrado, por exemplo, na indústria de confecções.

O MoldesView permite a verificar se ocorrem problemas na solução gerada pelo Gerador de Plano de Corte, como sobreposições de peças e/ou peças encaixadas fora da área estabelecida do tecido. Além disso, com a visualização das soluções, comprovou-se visualmente que as melhores soluções obtidas estavam de acordo com as regras para encaixe das peças.

As figuras seguintes mostram alguns padrões de corte. Pode-se visualizar as soluções através do MoldesView e tirar conclusões referentes aos encaixes.



Figura5 - Soluções MoldesView (Erro)

Verifica-se, através da Figura5, que a peça em destaque (oito) não se encontra no melhor posicionamento possível, ou seja, a melhor forma de aloca-la seria na lacuna existente entre as peças dois e sete.

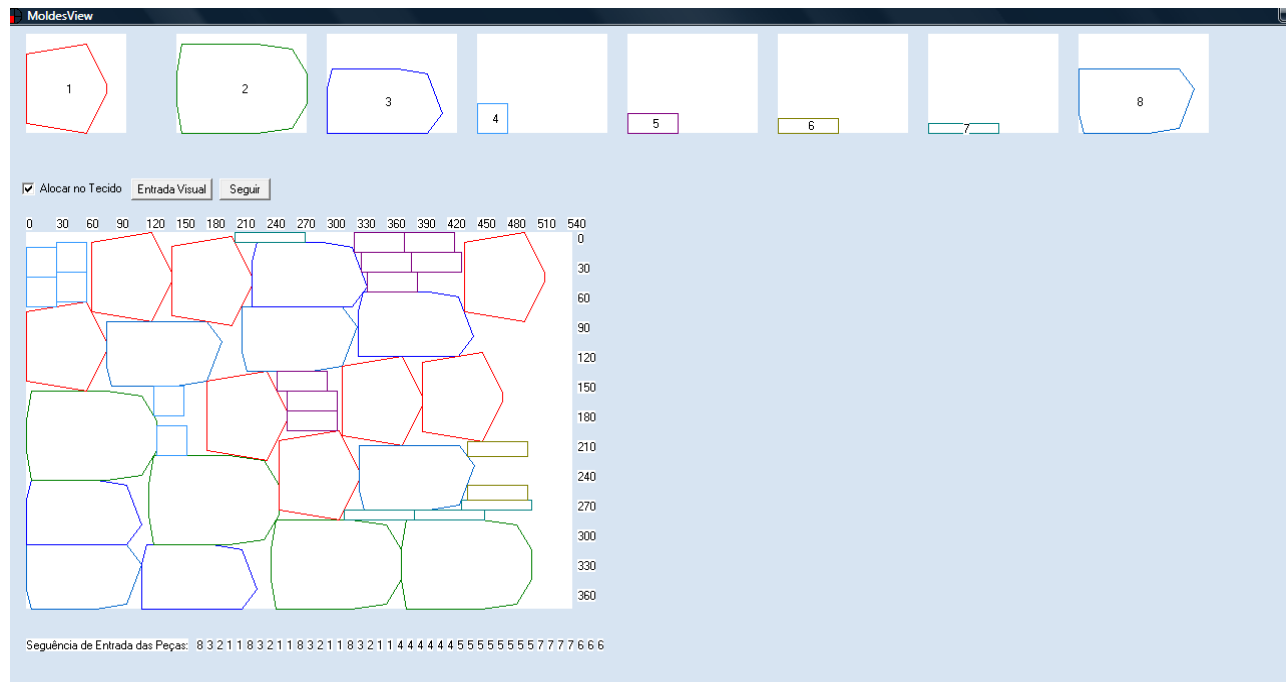


Figura6 - Encaixe de uma Solução Viável

Além da possibilidade de visualizar erros, a ferramenta visualiza uma solução viável, na qual todas as peças foram encaixadas de forma a minimizar a perda do tecido, essa visualização poder demonstrada na Figura6, acima.

5. CONCLUSÃO

A ferramenta desenvolvida viabiliza a visualização do posicionamento das peças que serão cortadas, ajudando, dessa forma, no desenvolvimento de padrões de corte eficientes, além de contribuir de forma direta com a redução do custo da matéria prima utilizada em confecções. Melhorando assim, o processo para a fabricação das roupas e conseqüentemente colocando o produto no mercado mais competitivo e com um preço aceitável.

Vale ainda lembrar que a ferramenta tem grande importância para seus usuários, tanto para aquele que apenas explora a visualização gráfica para depois cortar, quanto para aquele responsável pelo desenvolvimento de ferramentas com vistas à otimização de problemas de corte e empacotamento.

6. REFERÊNCIAS

CAMPOS, S. C., Sistema Gerador de Planos de Corte Guilhotinado e Bidimensional. Monografia, Ciência da Computação. Faculdade Ubaense Ozanam Coelho (FAGOC), Ubá, 2007.

FRANCESCHINELLI M., Problema de Corte – Heurísticas – Soluções Inteiras, 2000; encontrado em: http://www.scielo.br/scielo.php?pid=S0101-82006000300003&script=sci_arttext em: 15/04/2009.

LUNA, H.P.L. e GOLDBARG, C.M., *Otimização Combinatória e Programação Linear*. Editora Campus, Rio de Janeiro, 2005, 2a. Edição, 536 pp.

MORABITO. R. & BELLUZZO. (2007). Optimizing the Cutting of Wood fibre plates in the hardboard industry, *European Journal of Operational Research*, 183, 1406-1420